

# 7 PAS mathematical models to achieve alignment

---

chapter 2	CRE alignment state of the art and scientific gap
-----------	---

Preference-based Accommodation Strategy (PAS) design and decision approach						
Developing PAS		Testing PAS			Evaluating PAS	
chapter 3	chapter 4	chapter 5	chapter 6	chapter 7	chapter 8	
fifteen basic concepts and definitions from decision, design and management theory	2nd procedural rationality	steps			steps	
	3rd structural rationality		stake- holders & activities		stake- holders & activities	
	1st substan- tive ratio- nality			model	model	

chapter 9	Reflecting upon PAS
chapter 10	Conclusions and recommendations

# 7 PAS mathematical models to achieve alignment

---

The focus in this chapter is on the component mathematical models of PAS (see [Figure 7.1](#) and [Figure 7.2](#)). PAS can only be performed if the system engineers are able to build a mathematical model of the problem situation for each of the pilot studies. In this chapter, I will show that the system engineers were able to do this for all three pilots.

Typically, a subset of the alternative is infeasible. When the feasible set of alternatives can be characterized mathematically, the PFM algorithm can search an optimal alternative within this set (either by an exhaustive search or by sampling, depending on the size of the feasible set). Otherwise, if a characterization of the feasible set is not available to the algorithm, the group decision makers – the stakeholders – can propose possible feasible alternatives for consideration. The algorithm can then rate these alternatives.

This chapter has the following structure:

- TU Delft pilot for the food facilities in paragraph 7.1;
- TU Delft pilot for lecture halls in paragraph 7.2;
- Oracle's pilot for office locations in paragraph 7.3;
- Pilot comparison and conclusion in paragraph 7.4.

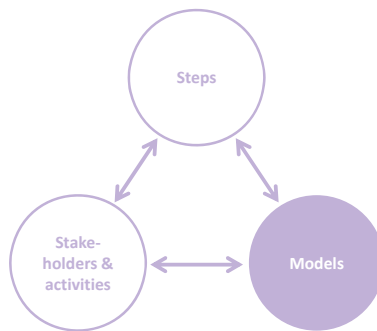
The mathematical models are explained for each of the pilots as follows: the model structure (first subparagraph), the model formulas (second subparagraph) and the optimization tool (third subparagraph).

Recall, that in step 5 alternatives are generated in two separate ways:

- A The group of decision makers self-designs alternatives, use the design constraints to test the feasibility of the design alternatives, and use the PFM algorithm to yield an overall preference score of these feasible design alternatives;
- B The system engineer generates feasible design alternatives and uses the PFM algorithm to find the feasible design alternative with the highest overall preference score.

The decision makers are able to design alternatives (step 5a) with the model that is explained in the first and second subparagraphs. The system engineer is able to generate alternatives (step 5b) with the optimization tool is, as is explained in the third subparagraph.

The mathematical models for the pilot studies have been built by the system engineer and the facilitator. The author had the role of the facilitator. The system engineer for the first pilot was Binnekamp, for the second pilot it was Valks with the aid of Barendse, and for the third pilot the system engineers were De Visser with the guidance of De Graaf. Valks and De Visser cooperated in this study as graduate students with the author as their main mentor and Binnekamp, Barendse and De Graaf as their second and/or third mentors.



**FIG. 7.1** Focus on PAS component mathematical model  
Note adapted from Arkesteijn et al. 2017, p. 245

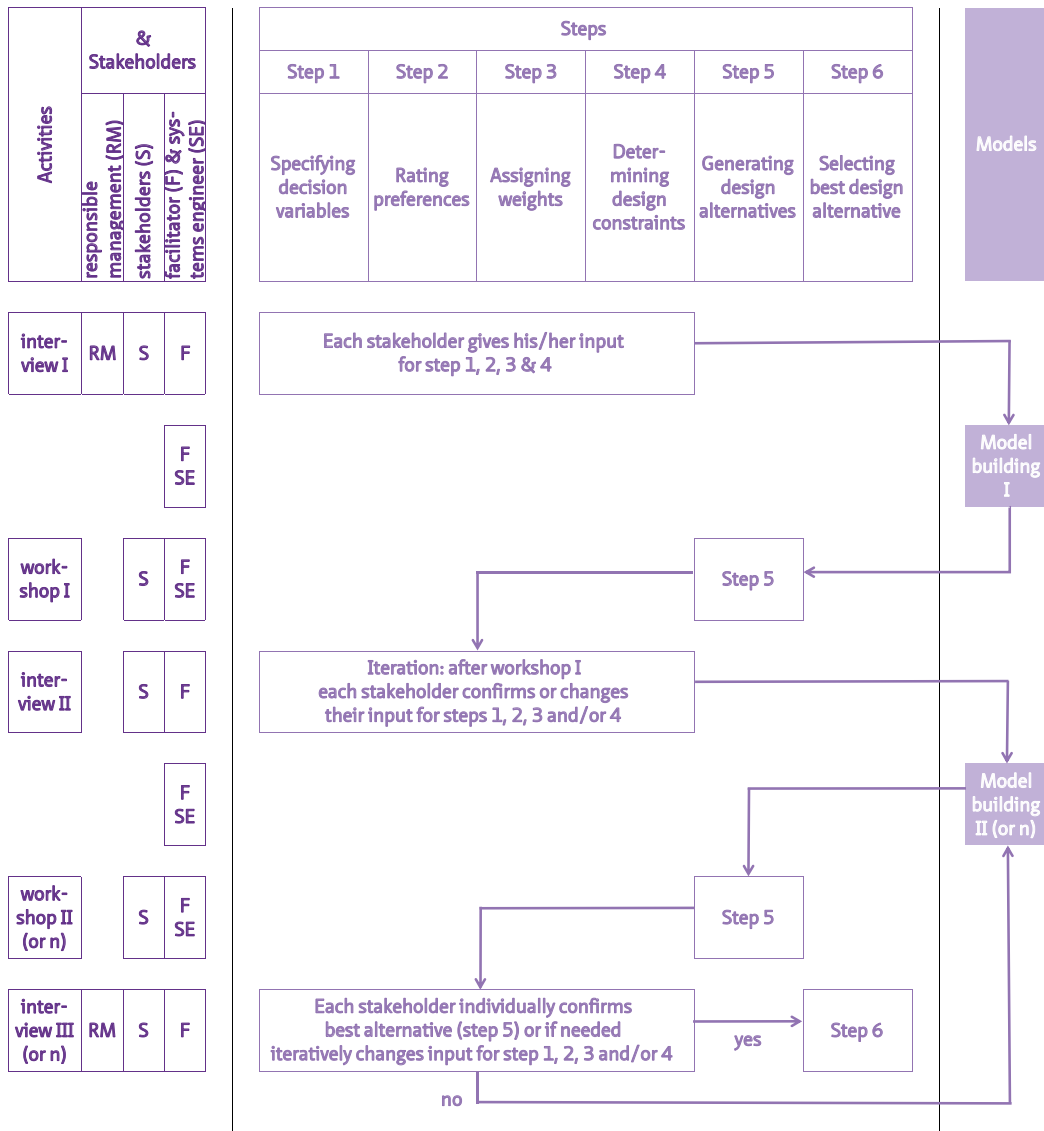


FIG. 7.2 PAS flowchart with emphasis on mathematical models Note adapted from Arksteijn et al. 2017, p. 248

## 7.1 Pilot study 1: TU Delft's food facilities

---

### 7.1.1 Model's structure

---

The structure of the mathematical model for the food facilities is similar to the generic structure of the mathematical model. In principal this generic mathematical model suffices when no special circumstances are present.

### 7.1.2 Model's Formulas

---

In this paragraph, the functions that computed the preference score of the decision variables are described. Firstly, an alternative as input is described, secondly the functions that calculate the *decision variable value* per decision variable and thirdly, the specific functions in this pilot that convert the decision variable into a preference score per decision variable.

#### An alternative as input

An *alternative* is described as follows:

The *state vector* is an alternative in the form  $(x_1, \dots, x_{16})$  where  $x_j$  is the state of food facility  $j$ .

At any given time, a dynamic system has a *state* given by a tuple of real numbers (a vector) that can be represented by a point in an appropriate state space.

The state of a food facility ( $x$ ) could be:

- 1 No food facility;
- 2 Coffee corner;
- 3 Restaurant for lunch without coffee corner;
- 4 Restaurant for lunch and dinner with coffee corner;
- 5 Restaurant for lunch with coffee corner;
- 6 *Faculty club*;
- 7 Restaurant concept *middle*;

- 8 Restaurant concept *large*;
- 9 Coffee corner with workplaces;
- 10 Restaurant for lunch without coffee corner with workplaces;
- 11 Restaurant for lunch and dinner with coffee corner with workplaces;
- 12 Restaurant for lunch with coffee corner with workplaces.

Note, that even though twelve states were available, not each state was feasible for each food facility (see [Figure 7.3](#)).

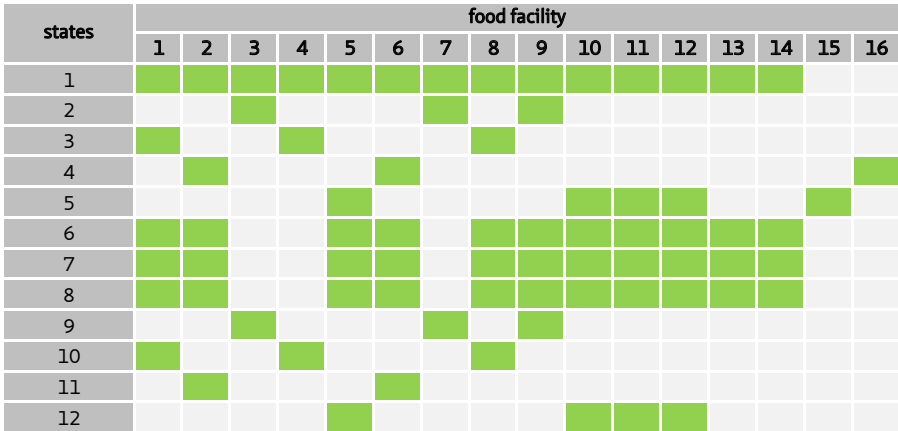


FIG. 7.3 Feasible states per food facility ( grey = infeasible and green = feasible)

To understand why not all states were feasible it is important to understand the difference between a state and an *intervention*. An intervention is a transformation that brings a system from one state into another. The interventions, as presented in paragraph 5.1, in this pilot were:

- 1 Refrain from action;
- 2 Remove the food facility;
- 3 Convert the existing food facility to new concept *middle*, *large* or *faculty club*;
- 4 Create a new concept *middle*, *large* or *faculty club*;
- 5 Upgrade the existing food facility.

Going back to the states, states 2 to 5 are related to the *current situation*, whereas states 1 and 6 to 12 are related to the *future situation*. In general, each current food facility can be transformed into state 6 to 8. Current food facility state 2 (coffee corner) can be transformed into state 9 (coffee corner with work places) with

intervention 5 (upgrade the existing food facility). The same holds for current food facilities with states 3, 4 and 5; they can be transformed respectively into state 10, 11 and 12 as can be seen in [Figure 7.3](#) . This means states 9 to 12 are the outcome of intervention 3 in a current food facility. Food facilities 15 (Sports Centre) and 16 (Inholland) cannot be transformed because they are not in ownership of TU Delft (see also [Table 5.1](#)).

In the mathematical model, as shown in chapter 6, the stakeholders were presented with feasible interventions instead of the abovementioned states.

**Functions to calculate the decision variable value per decision variable**

The model has the following twenty-nine functions.

In all functions the *state vector* is an alternative in the form  $(x_1, \dots, x_{16})$  where  $x_j$  is the state of food facility  $j$ .

If the *state vector* is known, the following eleven functions VAR\_1 to VAR\_11 obtained the decision variable value per decision variable from the dataset. The variables are not similar to the unique variables as explained in [Table 5.4](#).

*Var\_1*

Syntax                Var\_1(StateVector)

Return value        This function returns the value of the decision variable for decision variable 1 being the minimal walking time (in minutes) from a faculty to a food facility of concept 1 as it applies to users of all faculties. Concept 1 refers to the food facilities with state 3, 5, 7, 10 and 12.

*Var\_2*

Syntax                Var\_2(StateVector)

Return value        This function returns the value of the decision variable for decision variable 2 being the minimal walking time (in minutes) from a faculty to a food facility of concept 2 as it applies to users of all faculties. Concept refers to the food facilities with state 4, 8 and 11.



### *Var\_3*

Syntax            `Var_3(StateVector)`

Return value    This function returns the value of the decision variable for decision variable 3 being the percentage of seats in all food facilities that can be used for working.

### *Var\_4*

Syntax            `Var_4(StateVector)`

Return value    This function returns the value of the decision variable for decision variable 4 being the percentage of seats having a good acoustics over all food facilities.

### *Var\_5*

Syntax            `Var_5(StateVector)`

Return value    This function returns the value of the decision variable for decision variable 5 being the average ambience of all food facilities.

### *Var\_6*

Syntax            `Var_6(StateVector)`

Return value    This function the value of the decision variable for decision variable 6 being the average number of floors of where the food facilities are located.

### *Var\_7*

Syntax            `Var_7(StateVector)`

Return value    This function returns the value of the decision variable for decision variable 7 being the average accessibility of all food facilities, being the average number of doors in a faculty between its entrance and its food facility [doors].

### *Var\_8*

Syntax                      `Var_8(StateVector)`

Return value      This function returns the value of the decision variable for decision variable 8 being the average walking time (in minutes) from the entrance of a building to a food facility.

### *Var\_9*

Syntax                      `Var_9(StateVector)`

Return value      This function returns the value of the decision variable for decision variable 9 being the average percentage of food facilities labeled diverse.

### *Var\_10*

Syntax                      `Var_10(StateVector)`

Return value      This function returns the value of the decision variable for decision variable 10 being the average coziness of food facilities.

### *Var\_11*

Syntax                      `Var_11(StateVector)`

Return value      This function returns the value of the decision variable for decision variable 11 being the average findability of the food facilities.

## **Functions to convert the decision variable value into a preference score per decision variable**

Now that the decision variable values per decision variable are known, they can be converted into a preference score using the principal formula as described in paragraph 4.5.1. In the workspace, one of the variables is a column vector containing the weights of each decision variable given by the stakeholders corrected for stakeholder weight see paragraph 5.1. This variable is called `CritWeights`, see last column in [Table 7.1](#). The first two columns show the related decision variable and functions. The index refers to the unique preference score and is not related to earlier indices.

TABLE 7.1 Workspace pilot 1

Decision Variable	Function	Index	Weight
Minimal walking distance to concept 1	Pref_1_FS	1	0.0875
Minimal walking distance to concept 1	Pref_1_Staff	2	0.0250
Minimal walking distance to concept 1	Pref_1_Stud	3	0.0750
Minimal walking distance to concept 2	Pref_2_FS	4	0.0125
Minimal walking distance to concept 2	Pref_2_Staff	5	0.0625
Minimal walking distance to concept 2	Pref_2_Stud	6	0.0250
Percentage of restaurant seats usable as working place per restaurant seat	Pref_3_FS	7	0.0500
Percentage of restaurant seats usable as working place per restaurant seat	Pref_3_SI	8	0.1250
Percentage of restaurant seats usable as working place per restaurant seat	Pref_3_Stud	9	0.0375
Acoustics	Pref_4_FS	10	0.0500
Ambiance	Pref_5_FS	11	0.0500
Restaurant location (floors)	Pref_6_Stud	12	0.0500
Accessibility (number of doors)	Pref_7_Stud	13	0.0250
Walking distance within building	Pref_8_Stud	14	0.0375
Diversity	Pref_9_Staff	15	0.0625
Coziness	Pref_10_Staff	16	0.1000
Findability	Pref_11_SI	17	0.1250

If the *state vector* is known, the following seventeen functions convert the decision variable value per decision variable into a preference score.

*Pref\_1\_FS*

Syntax      Pref\_1\_FS(VariableValue)

Return value      Given a decision variable value, this function returns the corresponding preference score for decision variable 1 for decision maker Faculty Secretaries.

### *Pref\_1\_Staff*<sup>64</sup>

Syntax            `Pref_1_Staff(VariableValue)`

Return value    Given a decision variable value, this function returns the corresponding preference score for decision variable 1 for decision maker Staff.

### *Pref\_1\_Stud*

Syntax            `Pref_1_Stud(VariableValue)`

Return value    Given a decision variable value, this function returns the corresponding preference score for decision variable 1 for decision maker Students.

### *Pref\_2\_FS*

Syntax            `Pref_2_FS(VariableValue)`

Return value    Given a decision variable value, this function returns the corresponding preference score for decision variable 2 for decision maker Faculty Secretaries.

### *Pref\_2\_Staff*

Syntax            `Pref_2_Staff(VariableValue)`

Return value    Given a decision variable value, this function returns the corresponding preference score for decision variable 2 for decision maker Staff.

---

<sup>64</sup> Staff is referred to as works council in the other chapters

### *Pref\_2\_Stud*

Syntax            `Pref_2_Stud(VariableValue)`

Return value    Given a decision variable value, this function returns the corresponding preference score for decision variable 2 for decision maker Students.

### *Pref\_3\_FS*

Syntax            `Pref_3_FS(VariableValue)`

Return value    Given a decision variable value, this function returns the corresponding preference score for decision variable 3 for decision maker Faculty Secretaries.

### *Pref\_3\_SI*

Syntax            `Pref_3_SI(VariableValue)`

Return value    Given a decision variable value, this function returns the corresponding preference score for decision variable 3 for decision maker Project Manager Social Innovation.

### *Pref\_3\_Stud*

Syntax            `Pref_3_Stud(VariableValue)`

Return value    Given a decision variable value, this function returns the corresponding preference score for decision variable 3 for decision maker Students.

### *Pref\_4\_FS*

Syntax            `Pref_4_FS(VariableValue)`

Return value    Given a decision variable value, this function returns the corresponding preference score for decision variable 4 for decision maker Faculty Secretaries.

### *Pref\_5\_FS*

Syntax            Pref\_5\_FS(VariableValue)

Return value    Given a decision variable value, this function returns the corresponding preference score for decision variable 5 for decision maker Faculty Secretaries.

### *Pref\_6\_Stud*

Syntax            Pref\_6\_Stud(VariableValue)

Return value    Given a decision variable value, this function returns the corresponding preference score for decision variable 6 for decision maker Students.

### *Pref\_7\_Stud*

Syntax            Pref\_7\_Stud (VariableValue)

Return value    Given a decision variable value, this function returns the corresponding preference score for decision variable 7 for decision maker Students.

### *Pref\_8\_Stud*

Syntax            Pref\_8\_Stud (VariableValue)

Return value    Given a decision variable value, this function returns the corresponding preference score for decision variable 8 for decision maker Students.

### *Pref\_9\_Staff*

Syntax            Pref\_9\_Staff (VariableValue)

Return value    Given a decision variable value, this function returns the corresponding preference score for decision variable 9 for decision maker Staff.

### *Pref\_10\_Staff*

Syntax                Pref\_10\_Staff (VariableValue)

Return value        Given a decision variable value, this function returns the corresponding preference score for decision variable 10 for decision maker Staff.

### *Pref\_11\_SI*

Syntax                Pref\_11\_SI (VariableValue)

Return value        Given a decision variable value, this function returns the corresponding preference score for decision variable 11 for decision maker Project Manager Social Innovation.

The final function

### *IsFeasible*

Syntax                IsFeasible(StateVector)

Return value        This function returns the value 1 if the state vector meets all constraints, i.e. is feasible and 0 if the state vector does not meet all constraints.

The constraints are given in [Figure 5.20](#).

In appendix F the MatLab source is included for the decision variable Minimum walking as an example. In appendix G the functions are elaborated upon. At the end of this appendix a table is given showing the relation between the different numbers.

## 7.1.3 Optimization tool

---

After this pilot was finished, a search algorithm was developed by Barzilai in 2014. This algorithm has been tested on the pilot's data. Firstly, the search algorithm is explained, followed by the results of the search algorithm and lastly the evaluation of the results.

### The search algorithm

The search algorithm (Barzilai, to be published) searches for an optimum state vector, i.e. alternative, based on the overall preference score.

The search algorithm finds a local optimum. A local optimum is a solution that is better than any other feasible solutions in its immediate, or local, vicinity (Ragsdale 2008, p. 342). However, a given local optimal solution may not be the best possible solution, or global optimum, to a problem.

### Results of the search algorithm

The search algorithm generated six vector states, i.e. alternatives, with a substantially higher overall preference score than the alternative as designed by the decision makers (see [Table 7.2](#)).

Alternatives 1 and 2 scored higher than the other four alternatives and were the *local optimum*. In this table next to these six alternatives, two alternatives are presented which served as reference. Alternative 7 was the best alternative the stakeholders designed. This alternative has been given an overall preference score of 100 by Barzilai in this comparison. Alternative 8 on the other hand was an alternative with a very low overall preference score and this has been given an overall preference score of 0 by Barzilai in this comparison. Note that, Barzilai's overall preference scores therefore differ from the overall preference scores from chapter 5.

TABLE 7.2 Overall preference scores state vectors found by search algorithm

# state vector	overall preference score
1	1.417.909
2	1.417.909
3	1.414.039
4	1.407.461
5	1.395.782
6	1.395.782
7	1.000.000
8	0



Evaluation of the results

The six alternatives as found by the search algorithm (numbers 1 to 6) were evaluated by determining their feasibility. As explained in 7.1.2, an alternative is infeasible if a subset, i.e. one or more food facilities, has a state that is infeasible.

The *state vectors*, i.e. alternatives, are shown in the first column in [Figure 7.4](#) and the sixteen food facilities are depicted in the first two rows. A red cell indicates that the state of this food facility is infeasible. As can be seen, all six state vectors are infeasible.

The infeasibility of a state vector can be caused by different reasons. State vector 1, for instance, has seven food facilities (numbers 4, 6, 7, 10, 11, 15 and 16) with an infeasible state. For example, food facility 7 is the coffee corner in the library which in this alternative has a concept *large*; due to the size of the coffee corner it is impossible to accommodate a food facility of concept *large* here. Therefore, this is infeasible. In fact, only 73 combinations out the 192 (12 times 16) are possible (38 %) as is shown in [Figure 7.3](#).

# alternative	food facility															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1																
2																
3																
4																
5																
6																

FIG. 7.4 Feasibility of the state vector (red = infeasible and grey = feasible)

In contrary to the search algorithm, the stakeholders were only presented with the feasible interventions per food facility. Therefore, they only needed to check the feasibility of the alternative with respect to the constraints.

In this pilot, the algorithm (step 5b) was not able to generate a local optimum with a higher overall preference than the best alternative the decision makers designed (step 5a). The reason for this was that a subset of the alternatives was infeasible. The feasible set of alternatives could not be characterized mathematically. This means, that a characterization of the feasible set is not available to the algorithm. The feasible alternative made by the group decision makers is the best alternative.

## 7.2 Pilot study 2: TU Delft's lecture halls<sup>65</sup>

---

The model for this pilot used the generic set up and structure as discussed in paragraph 4.5. However, this model differed from the generic model in one way because of the type of problem that is addressed in this pilot. In paragraph 5.2 it has become clear that the lecture halls coped with the following four problems:

- 1 The current supply of lecture halls did not meet present-day requirements with regard to facilities and capacity;
- 2 The university started a new curriculum the year after the pilot, which led to a changing demand for lecture halls;
- 3 There were too few types of educational facilities to accommodate this changing demand;
- 4 The current supply was used ineffectively.

This means that the model established a relationship between the demand for educational space and the supply of lecture halls. In order to model this relationship PAS was not sufficient and two extra requirements needed to be met:

- 1 The model must be able to make a timetable based on the educational demands for a certain amount of lecture halls;
- 2 The user must be able to incorporate time constraints per activity in order to make the timetable representative.

Next to PAS linear programming (LP) was used to fulfil these requirements (Valks, 2016, p.53).

### 7.2.1 Model's structure

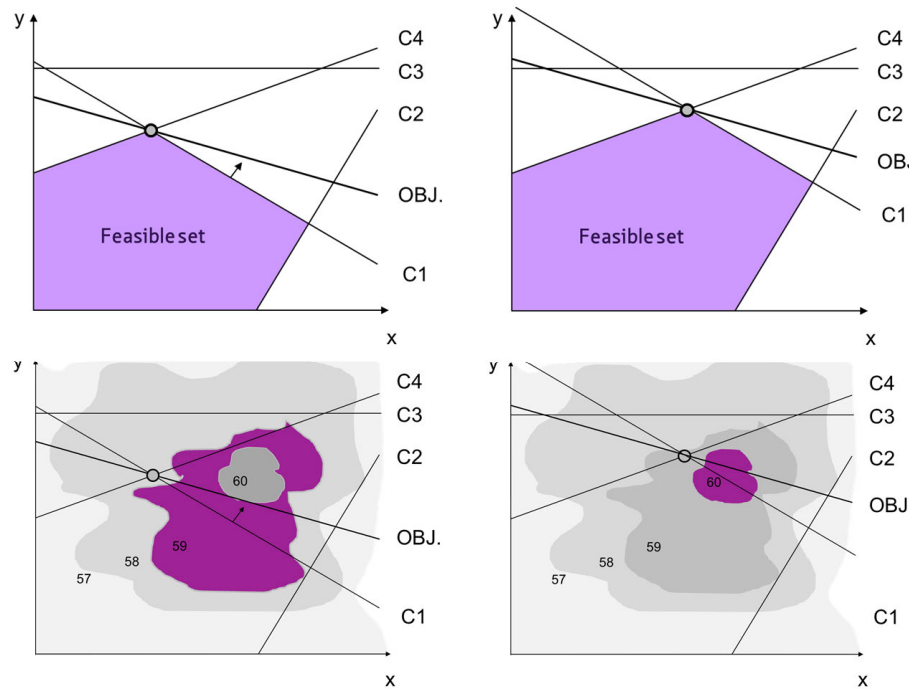
---

The model's structure is explained with a conceptual model. The conceptual design shows the relationship between PAS and the timetable allocation made by LP (see **Figure 7.5**) The key concepts for the conceptual model are the design space and the optimum alternative.

---

<sup>65</sup> This paragraph is based on Valks (2012, pp. 53– 55).

The primary objective in LP is to find a optimal timetable solution in the design space based on a single objective function. PAS is used to create a design alternative for the portfolio of lecture halls based on the preferences of stakeholders and to evaluate the timetable solution generated by LP. The primary objective in PAS is to design an optimum alternative with the highest overall preference score.



**FIG. 7.5** Models' structure; combining PAS and LP Note adapted from Valks, 2012, p. 54. In the two upper figures a constraint is modified to enlarge the set of feasible solutions; in the two lower figures the grey areas depict the overall preference score of the alternatives and enables the decision maker to choose the best one. The circle depicts the best alternative in LP given the objective (OBJ.).

In order to optimize the timetable allocation and thus achieve a higher overall preference score, the decision makers are able to influence the design space in two ways. First of all, the decision makers can modify a number of constraints that enlarge the design space, i.e. the feasible set of solutions. This means that the educational demands for a certain amount and/or type of lecture halls changes. It is than possible that this set has a design alternative with a higher preference score. Secondly the decision makers can alter the design space by designing an alternative in PAS. In this alternative the decision makers by choosing certain interventions have

changed the characteristics of the lecture halls and thus the supply of lecture halls. It is than possible that the design space has enlarged and that this set has a design alternative with a higher preference score (see **Figure 7.5**).

Different alternatives would have been chosen based on PAS and LP with a different overall preference score. Left top: circle with overall preference score of 58 and with PAS left bottom one of the alternatives in the feasible set and in the purple area with overall preference score 59. If the feasible set is enlarged the following happens, in the right top the circle with overall preference score of 59 and with PAS right bottom an alternative in the feasible set and in the purple area with overall preference score 60.

Because in this pilot it was impossible to design alternatives solely based on PAS, LP was needed to make a timetable based on the educational demands for a certain amount of lecture halls and to incorporate time constraints per activity. The timetable model in LP is subject to the same limitations as LP with negotiable constraints. However, within the timetable design space the decision makers are better equipped with PAS to select an alternative with a higher overall preference score (Valks, 2016).

---

### 7.2.2 Model's Formulas

The formulas of this model can be found in Valks (2013, pp. 54-59).

---

## 7.3 Pilot study 3: Oracle's office locations

The model for this pilot used the generic PAS model as discussed in paragraph 4.5. However, this model also differed from the generic model because an additional requirement was set. In order to compare PAS results with the original study, the model needed to calculate the preference score per decision variable for each of the locations and an overall preference score per location.

This paragraph is based on De Visser (2016, pp. 70-71, 76).

### 7.3.1 Model's structure

---

The model consisted of two parts:

- 1 Overall preference score design alternative;
- 2 Individual location preference scores.

Each of the parts is explained separately.

#### Overall preference score design alternative

The procedure to calculate the overall preference score used four steps and is shown in **Figure 7.6**. There was one input variable for the procedure that represents the design alternative. This variable was a list of all locations indicating respectively that a particular location was or was not included in the design alternative by the decision makers. In **Figure 7.6** this input is indicated by a X, this input is used in all functions. The functions are:

- 1 A function calculates the average design variable value for each decision variable (variable) based on the location data for this criterion;
- 2 A function calculates for each variable the preference score based on the design variable value and the preference points for this variable;
- 3 A function calculates the overall preference score for that particular portfolio design by combining the variable preferences score with the variable- and stakeholder weights;
- 4 A function tests the feasibility of the design alternative for all design constraints (De Visser 2016).

The output of the model is presented in the user interface (see paragraph 6.3).



### Preference score per location

This part of the model is additional to the generic model. It returns all individual design variable values and preference scores per design variable per location (see Figure 7.7). Based on these values, it calculates the overall preference score per location. The values and preference scores per design variable for each location remain the same in each workshop. Therefore, they were provided separately to the stakeholders to be used during the workshop generating alternatives (step 5). Next to that, a location ranking is produced that shows the locations and their overall preference score per location in descending order. This output is compared to the outcome of the initial study (see paragraph 5.3) (De Visser, 2016, pp. 70–71).

This model could be made because in this study all of the design variables were related to a single location. In the other pilots, design variables were formulated that set a requirement for the portfolio as a whole. In those pilots it is impossible to calculate the overall preference score for the underlying object.

---

#### 7.3.2 Model's Formulas

The formulas of this model can be found in De Visser (2016, pp. 118–122).

---

#### 7.3.3 Model's optimization tool

During the pilot study it became clear that this pilot was of such complexity that it was possible to generate all feasible design alternatives with a brute force approach. The complete pilot comprised of a total of 3.365.856 possible design alternatives. However, due to the design constraints a design space of only 4.480 feasible design alternatives remained. The alternatives could be generated by a brute force approach. With the use of a mathematical function the feasible alternatives were found and their overall preference score was calculated. The design alternative with the highest overall preference score is the global optimum (De Visser, 2016, p. 76).

## 7.4 Pilot study comparison and conclusion

---

In this chapter, it is shown that the system engineers were able to build mathematical models for each of the three pilots. Only in the second pilot, next to the PAS mathematical model, linear programming was needed as addition to incorporate time tabling. With these models, as has been shown in chapter 5, in each pilot, the decision makers were able to design (step 5a) and select (step 6) an *alternative* with a higher overall preference score than in the *current situation*. In each pilot more alignment have been achieved between CRE and the organization.

In addition to this, an optimization tool has been used in two pilots (step5b) with the aim to design an optimum alternative and achieve even more alignment. In the third Oracle pilot the optimization tool was successful. The optimization tool generated a *global optimum*. This means that design alternative (step 5b) has the highest overall preference score possible, i.e. also higher than the alternative the decision makers designed (step 5a). Due to the nature of the pilot the *brute force approach* could be used.

In first TU Delft pilot for the food facilities pilot, the algorithm (step 5b) was not able to generate a local optimum with a higher overall preference than the best alternative the decision makers designed (step 5a). The reason for this was that a subset of the alternatives was infeasible. The feasible set of alternatives could not be characterized mathematically and was not available to the algorithm. The feasible alternative made by the group decision makers is the best alternative.

In the second pilot, it was impossible to design alternatives solely based on PAS. Linear programming was needed to make a timetable based on the educational demands for a certain amount of lecture halls and to incorporate time constraints per activity. The timetable model in LP is subject to the same limitations as LP with negotiable constraints. However, within the timetable design space the decision makers are better equipped with PAS to design an alternative with a higher overall preference score.

In the third pilot, next to the overall preference score for the whole corporate real estate portfolio under investigation, it was also possible to calculate the overall preference score of the underlying object, i.e. a location. This gives the decision makers valuable extra design information. However, this is only possible if the decision makers do not formulate decision variables that set a requirement for the portfolio as a whole.